# MetacodeR: An R package for metabarcoding visualizations and primer evaluation

Zachary Foster and Niklaus Grunwald

May 17, 2016

## Introducing MetacodeR

Metabarcoding is revolutionizing microbial ecology and presenting new challenges:

- Numerous formats make taxonomic data difficult to manipulate.
- Stacked bar charts lack taxonomic context.
- Barcode loci and primers are a source of under-explored bias.

MetacodeR is an R package that attempts to addresses these issues:

- Taxonomic data can be extracted from any file format and manipulated.
- Community diversity can be visualized by color and size in a tree plot.
- Primer specificity can be estimated with *in silico* PCR.

## Parsing taxonomic data: Embedded classifications

The code below parses the Mothur 16s RDP training set.

```r
library(metacoder)
seqs <- ape::read.FASTA("trainset10_082014.rdp.fasta")
```

```r
cat(names(seqs)[1])
```

```
## AB294171_S001198039  Root;Bacteria;Firmicutes;Bacilli;Lactobacillales;Carnob
```

```r
data <- extract_taxonomy(seqs[1:1000],
                         regex = "^(.*)\\t(.*)",
                         key = c(rdp_id = "obs_info", "class"),
                         class_sep = ";")
```

```r
taxon_data(data, row_subset = 1:4)
```

```
## # A tibble: 4 x 9
##   taxon_ids supertaxon_ids            name n_obs n_obs_1 n_supertaxa
##       <chr>          <chr>           <chr> <dbl>   <dbl>       <dbl>
## 1         1           <NA>            Root  1000       0           0
## 2         2              1         Archaea    37       0           1
## 3         3              1        Bacteria   963       0           1
## 4         4              2 "Crenarchaeota"     6       0           2
## # ... with 3 more variables: n_subtaxa <dbl>, n_subtaxa_1 <dbl>,
## #   hierarchies <chr>
```

## Parsing taxonomic data: Genbank accession numbers

```r
ids <- c("JQ086376.1", "AM946981.2", "JQ182735.1", "CP001396.1", "J02459.1","AC
contaminants <- extract_taxonomy(ids, key = c("obs_id"),
                                 database = "ncbi")
```

```r
taxon_data(contaminants, row_subset = 1:4)
```

```
## # A tibble: 4 x 11
##   taxon_ids supertaxon_ids                name         rank ncbi_id n_obs
##       <chr>          <chr>               <chr>        <chr>   <dbl> <dbl>
## 1         1           <NA> cellular organisms      no rank  131567    59
## 2         2           <NA>    other sequences      no rank   28384    25
## 3         3           <NA>            Viruses superkingdom   10239    13
## 4         4              1           Bacteria superkingdom       2    56
## # ... with 5 more variables: n_obs_1 <dbl>, n_supertaxa <dbl>,
## #   n_subtaxa <dbl>, n_subtaxa_1 <dbl>, hierarchies <chr>
```

## Parsing taxonomic data: Taxon names

Parsing bryophyte family names scraped from The Plant List:

```
taxon_names <- "http://www.theplantlist.org/1.1/browse/B/" %>%
  XML::htmlTreeParse() %>%
  XML::xmlRoot() %>%
  XML::getNodeSet("//ul[@id='nametree']/li/a/i") %>%
  sapply(XML::xmlValue)
head(taxon_names)
```

```
## [1] "Acrobolbaceae"     "Adelanthaceae"     "Allisoniaceae"
## [4] "Amblystegiaceae"   "Anastrophyllaceae" "Andreaeaceae"
```

```
bryophytes_ex_data <- extract_taxonomy(taxon_names, key = "name",
                                        database = "itis")
```

```
taxon_data(bryophytes_ex_data, row_subset = 20:23)
```

```
## # A tibble: 4 x 11
##   taxon_ids supertaxon_ids            name   rank itis_id n_obs n_obs_1
##       <chr>          <chr>           <chr>  <chr>   <dbl> <dbl>   <dbl>
## 1        20             19   Chrysophyceae  class    1448     2       0
## 2        21             20  Ochromonadales  order    1451     1       1
## 3        22             20 Phaeothamniales  order    1689     1       1
## 4        23             21    Dinobryaceae family    1514     1       1
## # ... with 4 more variables: n_supertaxa <dbl>, n_subtaxa <dbl>,
## #   n_subtaxa_1 <dbl>, hierarchies <chr>
```

## Parsing taxonomic data: Taxon IDs

Parsing included example data from the ITS1 database:

```
file_path <- system.file("extdata", "its1_chytridiomycota_hmm.fasta",
                         package = "metacoder")
sequences <- ape::read.FASTA(file_path)
cat(names(sequences)[1])
```

```
## HQ191393_ITS1_HMM|uncultured Chytridiomycota|tax_id:175247|ITS1 located by H
```

```
its1_ex_data <- extract_taxonomy(sequences,
                                 regex = "^.*\\|(.*)\\|tax_id:(.*)\\|(.*)$",
                                 key = c(taxon_name = "taxon_info",
                                         "taxon_id", description = "obs_info"),
                                 database = "ncbi")
```

```
taxon_data(its1_ex_data, row_subset = 17:20)
```

```
## # A tibble: 4 x 12
##   taxon_ids supertaxon_ids          name    rank ncbi_id
##       <chr>          <chr>         <chr>   <chr>   <dbl>
## 1        17             16     africanum species  692697
## 2        18             16   californicum species   64516
## 3        19             10 Synchytriaceae  family  286113
## 4        20             19    Synchytrium   genus  286114
## # ... with 7 more variables: taxon_name <chr>, n_obs <dbl>, n_obs_1 <dbl>,
## #   n_supertaxa <dbl>, n_subtaxa <dbl>, n_subtaxa_1 <dbl>,
## #   hierarchies <chr>
```

## Accessing parsed data

```r
taxon_data(its1_ex_data, row_subset = 17:20)
```

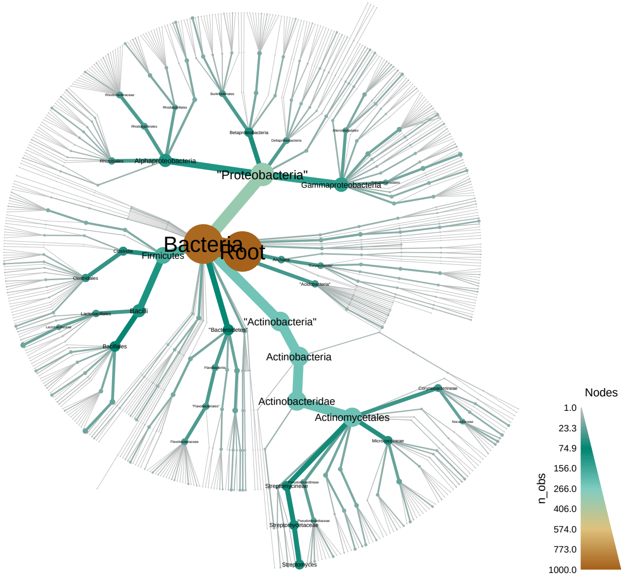
```
## # A tibble: 4 x 12
##   taxon_ids supertaxon_ids           name    rank ncbi_id
##       <chr>          <chr>          <chr>   <chr>   <dbl>
## 1        17             16      africanum species  692697
## 2        18             16    californicum species   64516
## 3        19             10 Synchytriaceae  family  286113
## 4        20             19    Synchytrium   genus  286114
## # ... with 7 more variables: taxon_name <chr>, n_obs <dbl>, n_obs_1 <dbl>,
## #   n_supertaxa <dbl>, n_subtaxa <dbl>, n_subtaxa_1 <dbl>,
## #   hierarchies <chr>
```

```r
obs_data(its1_ex_data, row_subset = 1:4)
```

```
## # A tibble: 4 x 4
##   obs_taxon_ids             obs_id
##           <chr>              <chr>
## 1           100 HQ191393_ITS1_HMM
## 2           100 HQ191391_ITS1_HMM
## 3            61 KJ464412_ITS1_HMM
## 4           100 HQ191291_ITS1_HMM
## # ... with 2 more variables: description <chr>, sequence <chr>
```

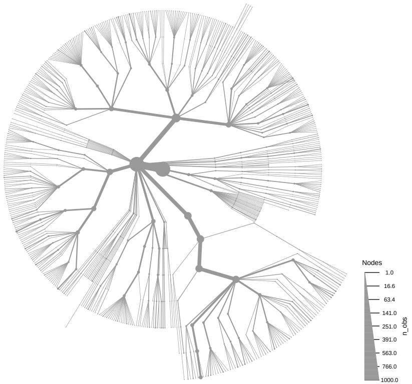
## Plotting taxonomic data: Metadiversity plots

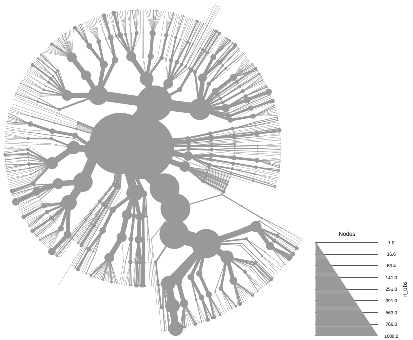
```
plot(data, node_size = n_obs,
     node_label = name, node_color = n_obs)
```



## Plotting taxonomic data: Overlap optimization

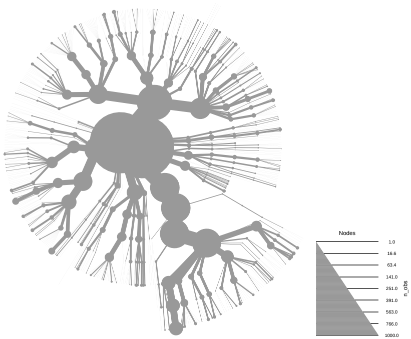
```r
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(data, node_size = n_obs, overlap_avoidance = 10),
  plot(data, node_size = n_obs, overlap_avoidance = 0.1))
```

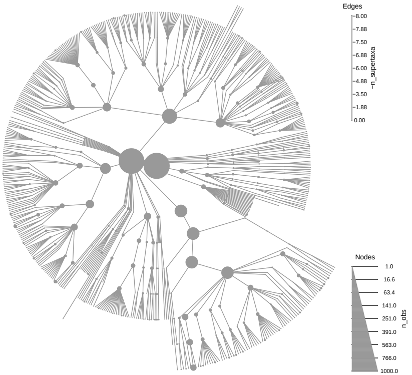




# Plotting taxonomic data: Size

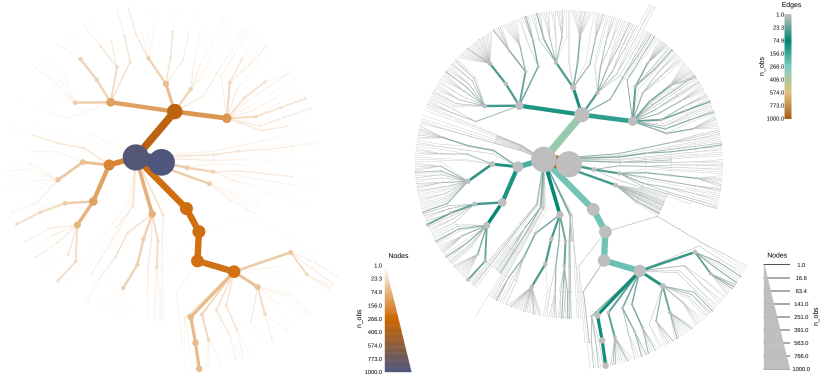
```r
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(data, node_size = n_obs,
       node_size_range = c(0.0001, 0.1)),
  plot(data, node_size = n_obs,
       edge_size = - n_supertaxa, edge_size_range = c(0.001, 0.001)))
```





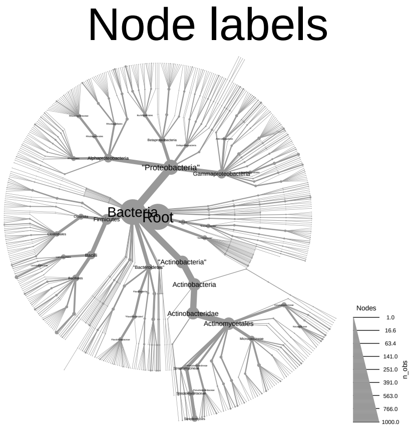
## Plotting taxonomic data: Color

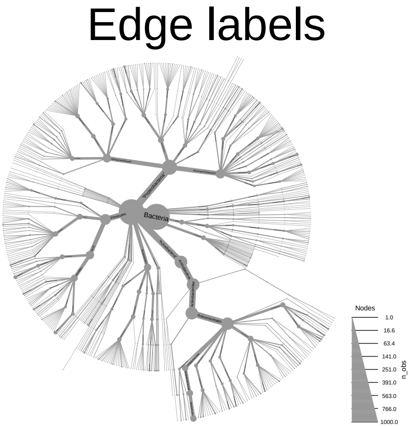
```
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(data, node_size = n_obs, node_color = n_obs,
       node_color_range = c("#FFFFFF", "darkorange3", "#4e567d")),
  plot(data, node_size = n_obs, node_color = "grey",
       edge_color = n_obs))
```



## Plotting taxonomic data: Labels

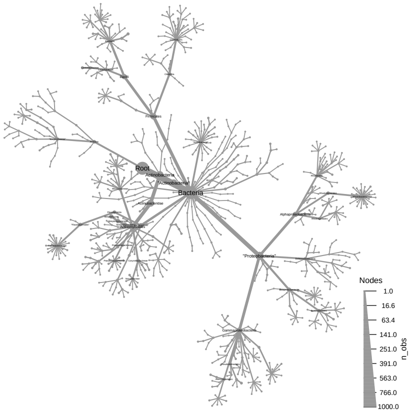
```r
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(data, node_size = n_obs, node_label = name,
       title = "Node labels"),
  plot(data, node_size = n_obs, edge_label = name,
       edge_label_max = 200, title = "Edge labels"))
```

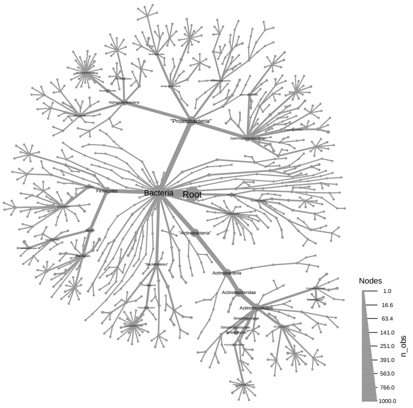




## Plotting taxonomic data: Layouts

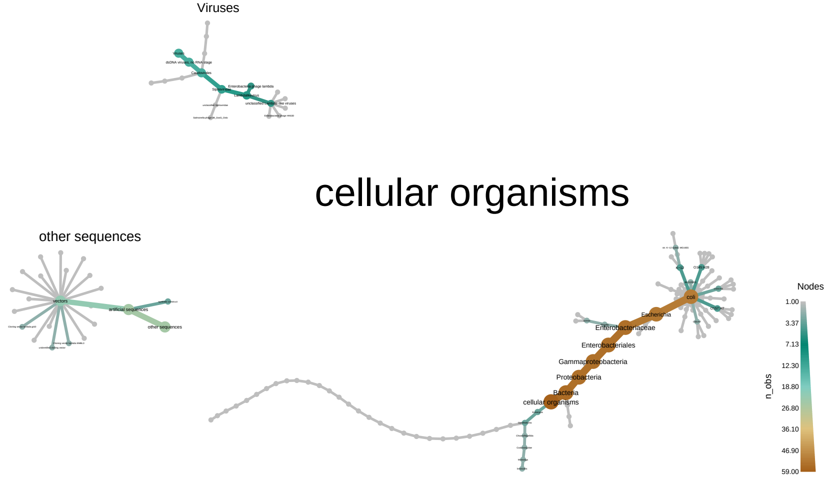
```r
set.seed(2)
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(data, node_size = n_obs, node_label = name,
       layout = "davidson-harel"),
  plot(data, node_size = n_obs, node_label = name,
       layout = "davidson-harel", initial_layout = "reingold"))
```





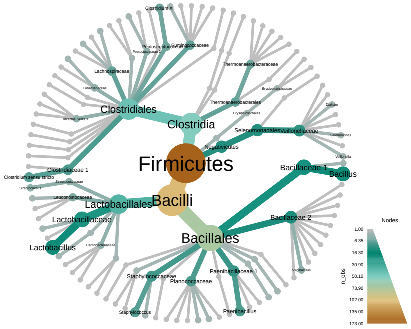
## Plotting taxonomic data: Multiple roots

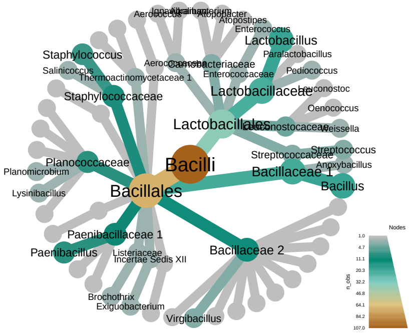
```r
set.seed(3)
plot(contaminants, node_size = n_obs,
     node_color = n_obs, node_label = name,
     tree_label = name, layout = "fruchterman-reingold")
```



## Subsetting taxonomic data: Picking a taxon

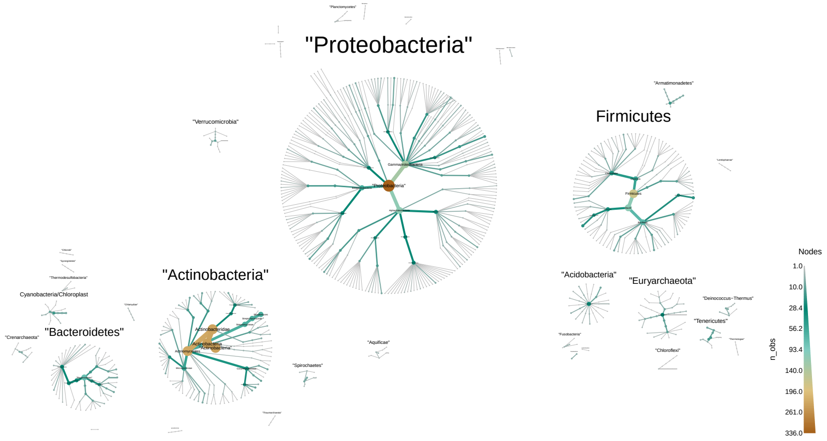
```r
set.seed(1)
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(filter_taxa(data, name == "Firmicutes", subtaxa = TRUE),
       node_size = n_obs, node_label = name,
       node_color = n_obs),
  plot(filter_taxa(data, name == "Bacilli", subtaxa = TRUE),
       node_size = n_obs, node_label = name,
       node_color = n_obs))
```





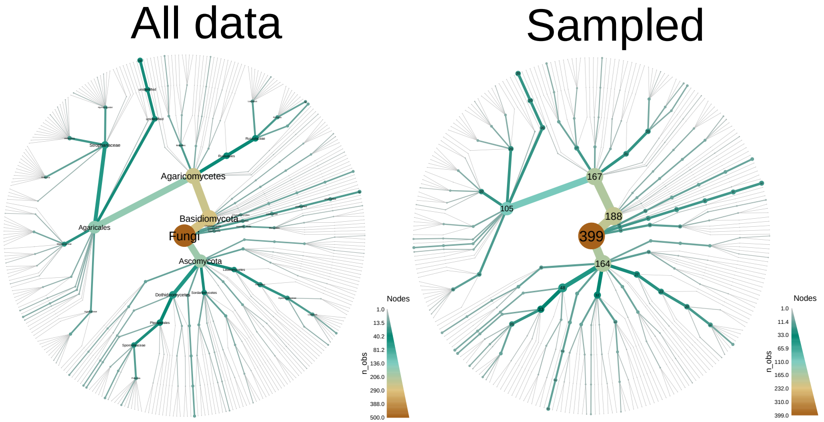
## Subsetting taxonomic data: Removing root taxa

```r
set.seed(1)
plot(filter_taxa(data, n_supertaxa > 1),
       node_size = n_obs, node_label = name,
       node_color = n_obs, tree_label = name)
```



## Sampling taxonomic data

```
subsampled <- taxonomic_sample(unite_ex_data_3, min_counts = c("7" = 3),
                               max_counts = c("4" = 20, "7" = 5))
gridExtra::grid.arrange(ncol = 2, nrow = 1,
  plot(unite_ex_data_3, node_size = n_obs, node_label = name,
       node_color = n_obs, title = "All data"),
  plot(filter_taxa(subsampled, n_obs > 0),
     node_size = n_obs, node_color = n_obs,
     node_label = n_obs, title = "Sampled"))
```



## *In silico PCR* use case example: Parsing data

```r
library(metacoder)
seqs <- seqinr::read.fasta("trainset14_032015.rdp.fasta")
```

```r
cat(names(seqs)[1])
```

```
## DQ343153_S000640727  Root;Bacteria;"Actinobacteria";Actinobacteria;Actinobac
```

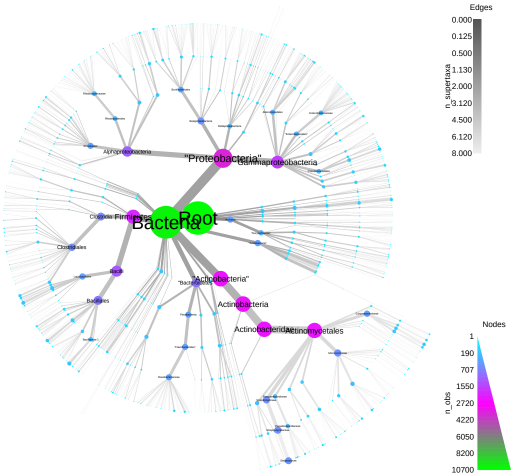
```r
data <- extract_taxonomy(seqs,
                         regex = "^>([a-zA-Z0-9_]+)[\t ]+(.*)$",
                         key = c(rdp_id = "obs_info", "class"),
                         class_sep = ";")
```

```r
taxon_data(data, row_subset = 1:4)
```

```
## # A tibble: 4 x 9
##   taxon_ids supertaxon_ids           name n_obs n_obs_1 n_supertaxa
##       <chr>          <chr>          <chr> <dbl>   <dbl>       <dbl>
## 1         1           <NA>           Root 10678       0           0
## 2         2              1        Archaea   434       0           1
## 3         3              1       Bacteria 10244       0           1
## 4         4              2 Aenigmarchaeota     1       1           2
## # ... with 3 more variables: n_subtaxa <dbl>, n_subtaxa_1 <dbl>,
## #   hierarchies <chr>
```

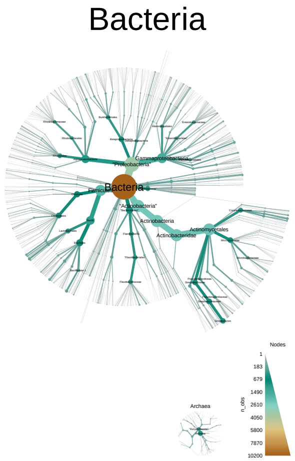
## *In silico* PCR use case example: Plotting

```
plot(data, node_size = n_obs, edge_color = n_supertaxa,
     node_label = name, node_color = n_obs,
     node_color_range = c("cyan", "magenta", "green"),
     edge_color_range   = c("#555555", "#EEEEEE"), overlap_avoidance = 0.5)
```



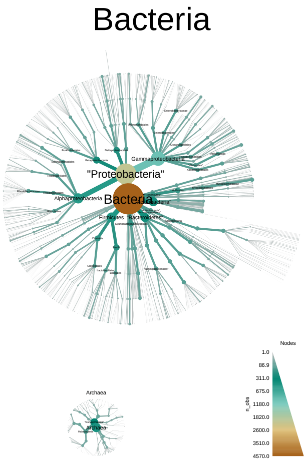
## *In silico* PCR use case example: Subsetting

```
subsetted <- filter_taxa(data, n_supertaxa > 0)
set.seed(2)
plot(subsetted, node_size = n_obs, node_label = name,
     node_color = n_obs, overlap_avoidance = 0.5, tree_label = name)
```



## *In silico* PCR use case example: Sampling

```r
sampled <- taxonomic_sample(subsetted, min_counts = c("6" = 5),
                            max_counts = c("3" = 100, "6" = 5))
sampled <- filter_taxa(sampled, n_obs > 0)
set.seed(3)
plot(sampled, node_size = n_obs, node_label = name,
     node_color = n_obs, overlap_avoidance = 0.5, tree_label = name)
```



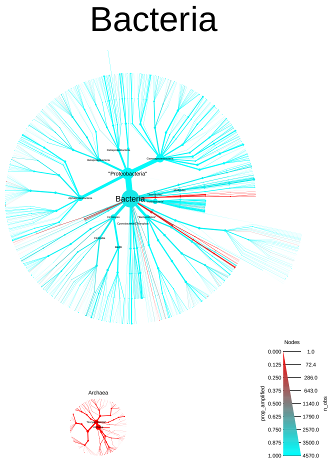
## *In silico* PCR use case example: First PCR

```r
pcr <- primersearch(sampled, forward = "CTCCTACGGGAGGCAGCAG",
                    reverse = "GWATTACCGCGGCKGCTG",
                    pair_name = "357F_519R",  mismatch = 10)
taxon_data(pcr, row_subset = 1:7)
```

```
## # A tibble: 7 x 11
##   taxon_ids supertaxon_ids             name n_obs n_obs_1 n_supertaxa
##       <chr>          <chr>            <chr> <dbl>   <dbl>       <dbl>
## 1         2           <NA>          Archaea   434       0           0
## 2         3           <NA>         Bacteria  4569       0           0
## 3         4              2  Aenigmarchaeota     1       1           1
## 4         5              2     Aigarchaeota     1       1           1
## 5         6              2  "Crenarchaeota"    55       0           1
## 6         7              2   Diapherotrites     2       2           1
## 7         8              2  "Euryarchaeota"    42       0           1
## # ... with 5 more variables: n_subtaxa <dbl>, n_subtaxa_1 <dbl>,
## #   hierarchies <chr>, count_amplified <dbl>, prop_amplified <dbl>
```

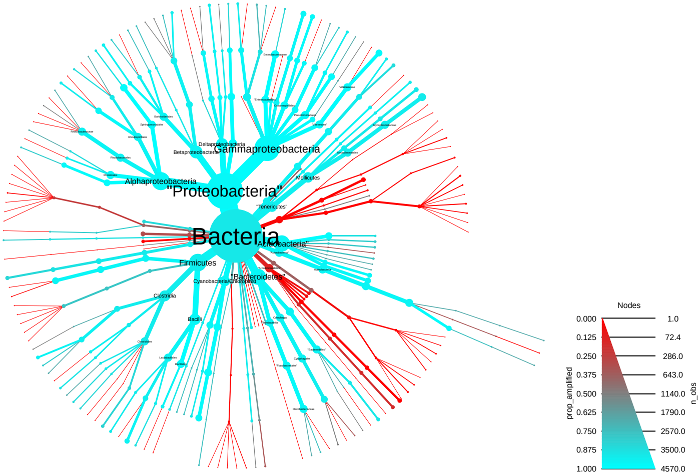
## *In silico* PCR use case example: First PCR

```r
set.seed(3)
plot(pcr, node_size = n_obs, node_label = name,
     node_color = prop_amplified, node_color_range =  c("red", "cyan"),
     node_color_trans = "linear", tree_label = name)
```



## *In silico* PCR use case example: First PCR

```
filter_taxa(pcr, name == "Bacteria", subtaxa = TRUE) %>%
  filter_taxa(count_amplified < n_obs) %>%
  plot(node_size = n_obs, node_label = name,
       node_color = prop_amplified, node_color_range =  c("red", "cyan"),
       node_color_interval = c(0, 1), node_color_trans = "linear")
```



## *In silico* PCR use case example: Second PCR

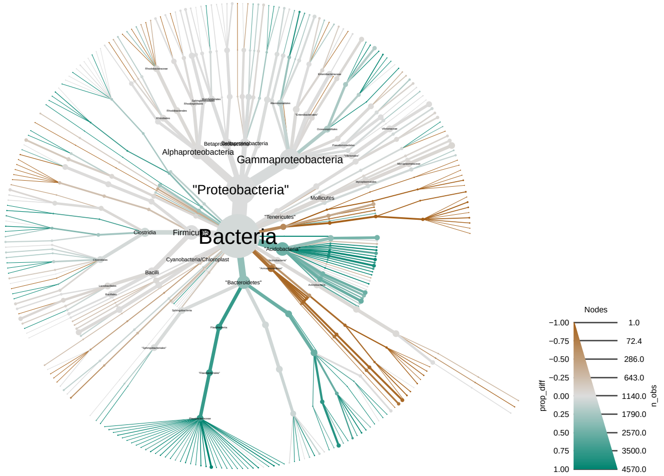
```
pcr_2 <- primersearch(sampled, forward = "GTGCCAGCMGCCGCGGTAA",
                      reverse = "AGGGTTGCGCTCGTTG",
                      pair_name = "515F_1100R", mismatch = 10)
taxon_data(pcr, row_subset = 1:7)
```

```
## # A tibble: 7 x 11
##   taxon_ids supertaxon_ids             name n_obs n_obs_1 n_supertaxa
##       <chr>          <chr>            <chr> <dbl>   <dbl>       <dbl>
## 1         2           <NA>          Archaea   434       0           0
## 2         3           <NA>         Bacteria  4569       0           0
## 3         4              2   Aenigmarchaeota     1       1           1
## 4         5              2      Aigarchaeota     1       1           1
## 5         6              2  "Crenarchaeota"    55       0           1
## 6         7              2   Diapherotrites     2       2           1
## 7         8              2  "Euryarchaeota"    42       0           1
## # ... with 5 more variables: n_subtaxa <dbl>, n_subtaxa_1 <dbl>,
## #   hierarchies <chr>, count_amplified <dbl>, prop_amplified <dbl>
```

```
pcr <- mutate_taxa(pcr,
                   count_amp_2 = taxon_data(pcr_2, col_subset = "count_amplifie
                   prop_diff = prop_amplified - taxon_data(pcr_2, col_subset = 
```

## *In silico* PCR use case example: Differential plot

```
filter_taxa(pcr, name == "Bacteria", subtaxa = TRUE) %>%
  filter_taxa(count_amplified < n_obs | count_amp_2 < n_obs) %>%
  plot(node_size = n_obs, node_label = name,
       node_color = prop_diff, node_color_range = diverging_palette(),
       node_color_interval = c(-1, 1), node_color_trans = "linear")
```



## Plans for future development

MetacodeR is under active development and many new features are planned. Some improvements that are being worked on include:

- Increases in function speed
- Plotting functions for pairwise comparison of treatments
- Barcoding gap analysis and associated plotting functions
- A function to aid in retrieving appropriate sequence data from NCBI for *in silico* PCR from whole genome sequences.

To see the details of what is being worked on, check out the issues tab of the MetacodeR Github site.